

DATABASE Expert

Never one to shy away from her studies, **Kay Ewbank** puts on her spectacles and considers a problematic bookshop database and an Access database for pupil assessment



NEW FILEMAKER RELEASE

FileMaker has released FileMaker Pro 8 and FileMaker Pro 8 Advanced. The new releases retain FileMaker's ease-of-use focus, but include features such as a PDF Maker that can be used to convert data reports into PDFs with graphics and layout information for non-FileMaker users. An Excel Maker provides similar facilities for Excel spreadsheets.

Other new facilities include a fast match facility that lets you select information from the current field and click to find all matching records. An auto-complete facility lets users enter data in fields automatically based on previous entries or Value Lists. There's also a visual spellchecker and improved email support. A new calendar field format gives the ability to show a date-picker for any field.

FileMaker Pro 8 Advanced is the new name for FileMaker Developer. Its new capabilities include custom menus and Tooltips. You can add, edit or delete menus and menu items. You can also execute scripts from custom menus as well as create custom scripts for toolbars, context menus and window options.

FileMaker has also announced the FileMaker Business Productivity Kit, which consists of several ready-to-use solutions designed for small businesses. The kit can be downloaded as a trial version from www.filemakertrial.com/bpkpr, as can the other versions.

FileMaker Pro 8 is available from www.filemaker.co.uk for £219 inc VAT for both FileMaker Pro 7 and six licensed users. FileMaker Pro 8 Advanced is £379 inc VAT.

FREE ACCESS ADD-ONS

Proximity Functions 1.1 Freeware is a set of free enhancements for Access databases. It includes Access modules, forms and controls that can be used to add features to your Access applications without programming. You import the modules into your application, then copy and paste the controls on to your own forms. Features include Windows-standard File Open/Save dialogs, folder browse dialogs, colour selection dialogs and a calendar date-picker dialog. See www.peterssoftware.com for more.

Q I have written a database for a local bookshop that has two networked computers. I have only used macros, and at 74 I am too old to start learning to code. The program works OK and does all that is needed for the running of the shop.

I want both users to have shared access and permission to make any changes they want. This all works well until I make a minor change to an object. After that the second user is told that they don't have permission to open the file. I'm using Access 2000, and the error message is:

"The Microsoft Jet database engine cannot open the file \\computername\shareddocs\filename.mdb. It is already opened exclusively by another user, or you need permission to view its data."

The same message appears whether the file on the other computer is open or closed. The Default Open Mode setting in Tools, Options, Advanced is Shared. In Tools, Security, User and Group Permissions, when I list users, the following permissions are ticked: Open/Run and Administer. Open Exclusive is unticked.

To correct the mysterious problem I have to make a new database, import everything except the switchboard items, then make a new switchboard. Please show me an easier way!
Bryan Mitchell

A There are several possible reasons why you might experience this problem. The most likely is that you need to set the correct network permissions for the directory in which you store your database.

When you open a database to modify objects in it, you have to open it exclusively. This creates a special file that has the same name as your main database file, but with an LDB extension rather than an MDB extension. This is the lock file that is used to determine which records are locked in a shared database, and who locked the records. It also contains a list of the users who have the database open. When a database is open for shared use, the LDB file is created and deleted automatically after the last user exits the database.

However, there are two exceptions to this. If the last user to exit the database doesn't have delete permissions on the folder

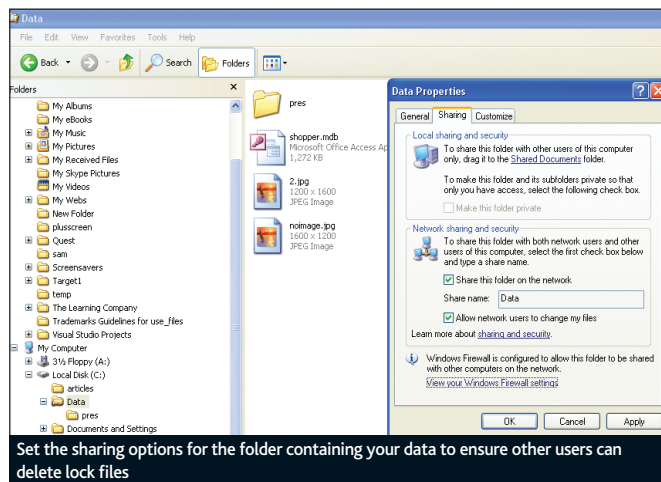
that contains the MDB file, or when the database is corrupted, then the lock file isn't deleted and this can cause problems later.

Your other user needs to have the permission in Windows ability to create, modify or delete this lock file. From Windows Explorer, right-click on the folder where your database is stored and choose Properties. You need to set the properties on the Sharing and Permissions tab to enable your other user to create, edit and delete files. You might find that the directory – or one of the directories that contains it – has a password set, in which case you can get around the problem by clearing the password.

While this is the most likely reason you're getting the problem, there are other possibilities. If you're using the Windows Accessibility options that are designed to make Windows easier to use for people who have poor eyesight, for example, Access can report the error that the database is locked by another user. This is a known error in Access 2000.

You can obtain a supported fix specifically for it from Microsoft Product Support Services. There is a list of phone numbers for Microsoft Product Support Services and information about support costs at the Microsoft website at <http://support.microsoft.com/default.aspx?scid=fh;EN-US;CNTACTMS>. For the fix to be applied, you will already need to have installed Office 2000 Service Pack 3 (SP3).

If you're not using Accessibility options, and the first suggestion didn't work, the other possibility is that the database has become damaged such that the error is triggered when you make a change. To get around this problem, try the Compact and Repair Utility. As its name suggests, this gets rid of empty space (the Compact) in a database. It can also be used to repair problems in an existing database. It only repairs the tables, queries and



Set the sharing options for the folder containing your data to ensure other users can delete lock files



indices in the database and not the forms, reports, macros and modules. However, it's still worth trying as it might solve the problem.

To repair the database, first you need to close it. Then choose Database Utilities from the Tools menu, and from there choose Compact and Repair Database. When you see the Compact Database From dialog box, click the file you want to compact and then click Compact. You'll be asked for a new filename for the repaired copy of the database.

If there's damage in one of your forms or reports, your only option is to recreate the damaged form or report, unless you have a backup copy of the database with an undamaged version of the form.

Q I am in the process of designing an Access database for pupil assessment and I would like to include a photograph of each pupil on their individual records. The photographs are in JPG format. I know I can resize, copy and paste each photograph manually, but is there a way to speed up the process rather than spending hours doing this with hundreds of photographs?

Bill Shaw

A You can certainly automate much of the work, but Access may not be the best place to start. I don't know which image-editing software you use, but some programs have batch options that let you make the same changes to a set of images, so you could do your resizing operation automatically. The excellent Photoshop Elements 3.0 offers batch resizing, and is our favourite budget image editor at around £50.

Pricier editors often support scripting, so you can write little macros to do work for you. There are also several free or inexpensive image-processing packages that claim to be able to make changes to multiple files in a batch operation. Type "batch image resize" into Google for several alternatives.

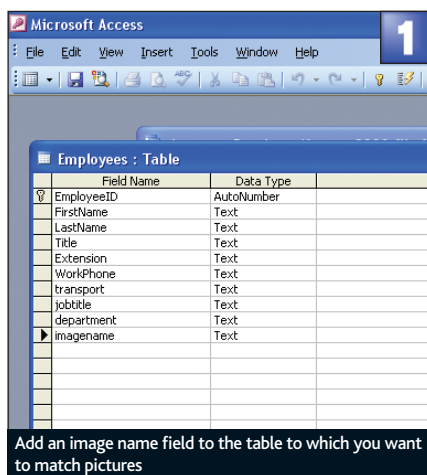
Once your image files are the right size, you have various choices in how to handle them. You don't have to bring them into Access. You could use a hyperlink field that contains the name and the path where the files are stored, which would require your users to click on a hyperlink to view the image.

If you're using the more usual method of an OLE Object field, you can still link the files and have them stored externally. You also have the option of embedding them, which I suspect is the route you've chosen from the description you give. While you'd probably be fine with the few records you have in your database, OLE fields are notorious for taking lots of space, so it isn't a good idea to embed the images. For example, a JPG of 50K will occupy around a megabyte of space in your database because of the way OLE objects are stored.

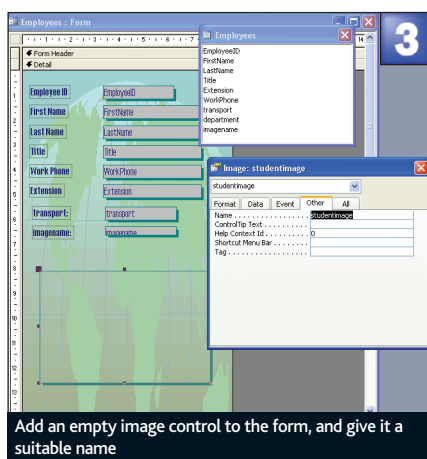
The neatest way to achieve what you want without getting an enormous database is to create an unbound image frame on your form and have a text field in your table holding the full name (including the path) of the pupil's digital photo file.

Add the following code to the OnCurrent event of your form. The code assumes that your unbound image frame is called StudentImage and the field in your table is called ImageName:

```
Private Sub Form_Current()  
    If IsNull(Me![ImageName]) Then  
        Me![StudentImage].Picture = "c:
```



Add an image name field to the table to which you want to match pictures



Add an empty image control to the form, and give it a suitable name

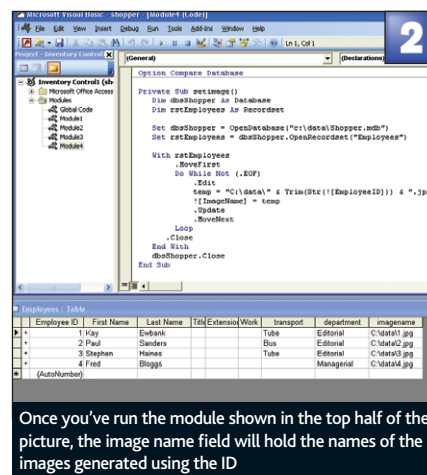
```
\\data\\noimage.jpg"  
Else  
    Me![StudentImage].Picture = Me!  
    ImageName  
End If  
End Sub
```

The first part of this code checks to see whether the ImageName field has a value entered in it. If it doesn't, a standard image called noimage.jpg is loaded; you'll need to create this and make sure it's stored in the correct location. Our database was stored in the C:\data folder, so that's where we stored noimage.jpg.

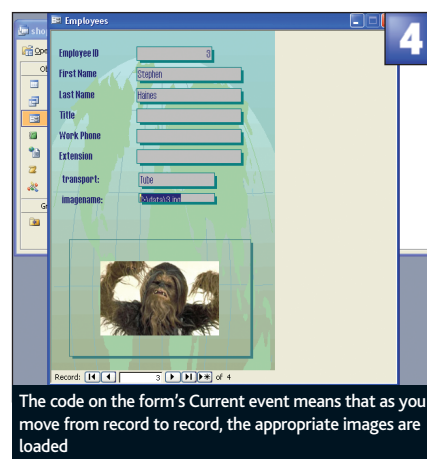
If the image name field has a value in it, that file is loaded and displayed in the unbound image frame. There will be a small time lag while the image loads, but it's only half a second or so. This method overcomes the need to import multiple images into the actual database; just set the field values for the image filename. The easiest way to match up student record and student image is to give each student a student number, and use that number as the name of the image. Student Fred Bloggs, with student number 2156, would have his image stored as 2156.jpg.

If you've named the images in this way, you can use a one-off VBA procedure along the following lines:

```
Private Sub setimage()  
    Dim dbsShopper As Database  
    Dim rstEmployees As Recordset  
    Set dbsShopper = OpenDatabase("c:
```



Once you've run the module shown in the top half of the picture, the image name field will hold the names of the images generated using the ID



The code on the form's Current event means that as you move from record to record, the appropriate images are loaded

```
\\data\\Shopper.mdb")  
Set rstEmployees = dbsShopper.  
OpenRecordset("Students")  
With rstEmployees  
    .MoveFirst  
    Do While Not (.EOF)  
        .Edit  
        temp = "C:\\data\\" & Trim(Str(!  
            [Studentid])) & ".jpg"  
        ![imagename] = temp  
        .Update  
        .MoveNext  
    Loop  
    .Close  
End With  
dbsShopper.Close  
End Sub
```

Run this procedure, changing the table names and field names to suit your particular database. You should find that all the image name fields have been set to something like c:\data\2156.jpg. You can then use the form with the empty image control, and when you move from record to record you'll find the images are loaded as required. **CS**

CONTACT

KAY EWBank

Kay is Computer Shopper's database expert and has worked as a consultant and author for 20 years

kay@computershopper.co.uk